UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

ELEKTRONICKÝ ZABEZPEČOVACÍ SYSTÉM NA BÁZI RASPBERRY PI

Dominik Mošner

Bakalářská práce

2019

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2018/2019**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Dominik Mošner**
Osobní číslo:	**I15018**
Studijní program:	**B2612 Elektrotechnika a informatika**
Studijní obor:	**Řízení procesů**
Název tématu:	**Elektronický zabezpečovací systém na bázi Raspberry Pi**
Zadávající katedra:	**Katedra řízení procesů**

Zásady pro vypracování:

Cíl: Cílem práce je realizace elektronického zabezpečovacího systému řízeného mikropočítačem Raspberry Pi. Systém bude schopen rozpoznávat pohyb a přenášet živý video záznam pomocí kamerového modulu. Systém bude dále schopen rozpoznat otevření dveří a tuto informaci předat uživateli prostřednictvím webové stránky nebo SMS či e-mailu.
Teoretická část: Stručná rešerše problematiky týkající se realizace nejčastěji používaných zabezpečovacích systémů a programování mikropočítače Raspberry Pi.
Implementační část: Tvorba a ověření aplikace pro mikropočítač. Dokumentace k řešení a popis návrhu řídicích a ovládacích algoritmů, diskuse k hardwarovému provedení, vyhodnocení funkčnosti a uživatelské přívětivosti.

Rozsah grafických prací:

Rozsah pracovní zprávy: **cca 50 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

UPTON, E.; HALFACREE, G. 2016. Raspberry Pi User Guide. 4th Ed. Chichester, West Sussex (UK): John Wiley & Sons. ISBN 978-1-119-26436-1.
ROBINSON, A.; COOK, M. 2014. Raspberry Pi: Projects. Chichester, West Sussex (UK): John Wiley & Sons. ISBN 978-1-118-55543-9.
MCMANUS, S.; COOK, M. 2013. Raspberry Pi for Dummies. 3rd Ed. Hoboken, New Jersey: John Wiley & Sons. ISBN 978-1-118-55421-0.

Vedoucí bakalářské práce: **Ing. Libor Kupka, Ph.D.**
Katedra řízení procesů

Datum zadání bakalářské práce: **14. prosince 2018**

Termín odevzdání bakalářské práce: **10. května 2019**

Univerzita Pardubice Česká republika

L.S.

Ing. Zdeněk Němec, Ph.D.
děkan

Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 14. prosince 2018

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 10. 5. 2019

Dominik Mošner

ANOTACE

Bakalářská práce je věnována návrhu zabezpečovacího systému řízeného mikropočítačem Raspberry Pi. V práci je vysvětlena problematika zabezpečovacích systémů a programování algoritmů v mikropočítači Raspberry Pi. Systém má implementován dva bezpečnostní prvky – rozeznání člověka pomocí kamerového modulu a rozpoznání otevření dveří.

KLÍČOVÁ SLOVA

Raspberry Pi, zabezpečovací systém, OpenCV, kamerový modul.

TITLE

RASPBERRY PI BASED SECURITY SYSTEM

ANNOTATION

Bachelor thesis is devoted to the implementation of a security system controlled by Raspberry Pi. The thesis deals with an implementation of security systems and programming algorithms in Raspberry Pi single-board computer. The system has implemented two security features – recognizing a person using camera module and recognizing a door opening.

KEYWORDS

Raspberry Pi, Security system, OpenCV, Camera module

OBSAH

SEZNAM ZKRATEK A ZNAČEK

CCD	Charge Coupled Device
CCT	Closed circuit television
CSI	Camera Serial Interface
DVR	Digital Video recorder
DSP	Digital Signal Processor
FTP	File Transfer Protocol
GPIO	General Purpose Input/Output
HDMI	High-Definition Multimedia Interface
I/O	Input/Ouput
IP	Internet Protocol
IMAP	Internet Message Access Protocol
MIME	Multipurpose Internet Mail Extensions
PoE	Power over Interent
POP3	Post Office Protocol
RGB	Red-Green-Blue
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
URL	Uniform Resource Locator
Wi-Fi	Wireless Fidelity
USB	Universal Serial Bus

SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ

$\boldsymbol{H}$	konvoluční maska
R	elektrický odpor, Ω
r	vzdálenost od souřadnicového systému, m
T	hodnota prahu
U	elektrické napětí, V
θ	úhel svírající přímka r s osou x, °
φ	úhel směru největší změny jasové funkce, °
$f(x, y)$	obrazová funkce
X	množina obrazu
B	množina masky
d	bod v obraze
E^2	binární obrazový prostor
x, y	kartézské souřadnice bodu

SEZNAM ILUSTRACÍ

SEZNAM TABULEK

ÚVOD

Zabezpečovací systémy jsou nedílnou součástí cenných objektů. Strmý vývoj informačních technologií způsobuje větší sofistikovanost zabezpečovacích zařízení. Nejčastějšími způsoby pro zabezpečení objektu je ostraha, použití mechanických zábranných systémů a v neposlední řadě také elektronické bezpečnostní systémy. V minulosti musela stráž celý den sledovat živý záznam z bezpečnostní kamery. V současnosti můžou být bezpečnostní systémy řízeny vzdáleně pomocí chytrého mobilního telefonu nebo dokáží sami upozornit uživatele na blížící se hrozbu. V dnešní době se nejvíce využívají bezpečností kamerové systémy typu CCTV a IP. Tyto systémy jsou velice účinné, ale náklady na implementaci se ukázaly být překážkou zejména pro malou domácí aplikaci. Proto v této práci je použit cenově dostupnější jednočipový počítač Raspberry Pi. Jedná se o velice efektivní řídicí jednotku, která může být propojena s dalšími moduly pro realizaci systémů s obrovskou rozmanitostí funkcí. Mikropočítač Raspberry Pi je možné použít na realizaci cenově výhodnějšího bezpečnostního systému, který se funkčností vyrovná profesionálnímu zabezpečovacímu zařízení. Tato práce se zabývá realizací takového zařízení.

1 ZABEZPEČOVACÍ SYSTÉMY

Zabezpečovací systémy se využívají k ochraně majetku, informací a osobní bezpečnosti. Úkolem chytrého bezpečného systému je detekovat jakýkoliv pokus o vniknutí a následně na něj upozornit a tím minimalizovat možné hmotné škody. Technická vybavenost zabezpečovacích systémů se v posledních letech prudce zlepšuje. Stojí za tím vývoj elektronických systémů a stále vyšší nároky uživatelů na kvalitu zabezpečení.

Kamerové systémy patří k nejpoužívanějšímu prvku zabezpečovacího systému. Při využití monitorovacího bezpečnostního prvku je naším cílem v první řadě odradit pachatele od zamýšleného činu. Pokud monitorovací prvek neodradí pachatele, systém se postará o to, aby jeho čin byl dobře zdokumentován. Poté už nic nebrání k tomu, aby policie vyhledala a potrestala pachatele.

1.1 KAMERY TYPU CCTV

Zkratka CCTV znamená „Closed Circuit Television“, přeloženo do českého jazyka jako „uzavřený televizní okruh“. CCTV videokamery slouží pro přenos videozáznamů na určité místo a na omezenou sadu monitorů. CCTV je starší bezpečností systém, který se vyvíjí a modernizuje po celé roky. Pokročilejší forma CCTV, využívající DVR, umožňuje nahrávání s různými kvalitami obrazu a dalšími funkcemi (například detekce pohybu a upozornění na email). Pokud je DVR připojen k internetu, může být také k dispozici vzdálené sledování videozáznamu. CCTV kamery se nejčastěji používají pro sledování velkých ploch, jako jsou maloobchodní prodejny, banky a další instituce.

Kamera zachytí analogový signál, který je převeden do digitálního videorekordéru. DVR konvertuje analogový signál na digitální, komprimuje ho a uloží na pevný disk. Snímek pak lze prohlížet na monitorech připojených k DVR nebo může být odeslán pomocí internetu.

Hlavní funkcí kamery CCTV je zachytit světlo a přeměnit ho na video signál. V jádru CCTV kamery je snímač CCD, který přeměňuje světlo na elektrický signál. Dále dojde k zpracování elektrického signálu a následnou přeměnu na video signál, který lze zobrazit na obrazovce nebo uložit na datové médium. CCD se skládá z polovodičových fotodiod. Objektiv kamery zaostřuje na snímač CCD. Fotodiody snímají oblasti obrazu světla a tmy a výsledkem je elektrický náboj v poměru k úrovni světla. Jasnější oblast způsobí vyšší náboj. Fotodiody tvoří matici řádků a sloupců a nazývají se obrazovými buňkami nebo pixely (Techcude Limited, 2016).

1.2 DIGITÁLNÍ KAMERY TYPU IP

Digitální kamery typu IP zachytí analogový signál, který je uvnitř samotné kamery přeměněn na digitální. Kamera se chová jako síťové zařízení, což znamená, že má přidělenou IP adresu. Kamera je opatřena vestavěným softwarem pro web server, FTP server a emailového klienta. Dále je opatřena jedním nebo více logickými vstupy a výstupy. Výstupy a vstupy slouží ke komunikaci s externími zařízeními, jako jsou například elektronicky ovládané brány, blokovací zámky či jiné funkcionality inteligentních budov. Řídicí procesor se stará o komunikaci se sítí, webovým serverem nebo ovládá externí zařízení. Síťový signál je přenášen pomocí pětivodičového kabelu Ethernet. A tímto způsobem je vedena komunikace v obou směrech.

Digitální kamery pro zpracování signálu využívají čip DSP. Analogový signál je generován čipem CCD a poté čip DSP převádí signál na digitální. Výhody kamer využívající čip DSP zahrnují zvýšenou jasnost, větší stabilitu obrazu, ostřejší obraz a lepší energetickou účinnost.

1.2.1 DSP

Čip DSP je založen na skutečnosti, že je možné sestavit reprezentaci analogového signálu v digitální podobě. To se provádí vzorkováním hladiny napětí v pravidelných časových intervalech a přeměnou na digitální číslo úměrné vzorkovanému napětí. Tento proces je prováděn analogově digitálním převodníkem (A/D převodník). DSP má mnoho výhod oproti analogovému zpracování. Je schopen poskytnout daleko lepší úroveň zpracování signálu, než je možné pouze s analogovým hardwarem. Je schopen provádět matematické operace, které umožňují filtrovat mnoho falešných signálů způsobující analogové komponenty. DSP není schopno zajistit dokonalou přeměnu signálu. Počítače nemají nekonečnou paměť a ani nejsou nekonečně rychlé, proto se A/D převodník musí omezit pouze na vzorkování určitého počtu úrovní signálu (Techcude Limited, 2016).

1.2.2 PoE

Kamery s technologií PoE umožňují napájení kamer z jediného síťového kabelu. Standardně kamera využívá dva kabely, jeden pro video signál a jeden pro napájení. S funkcí PoE je vyžadován pouze jeden kabel, který napájí a také přenáší video signál. Další typ přenosu signálu je pomocí bezdrátové sítě Wi-Fi. Bezdrátový přenos může být ovlivněn ztrátou spojení nebo slabým signálem. Bezdrátové kamery jsou oblíbené pro použití v domácnosti.

2 POČÍTAČOVÉ VIDĚNÍ

Počítačové vidění je disciplína, která se snaží technickými prostředky napodobit lidské vidění. Počítačové vidění má dvojí cíl. Z pohledu biologické vědy je navrhnout výpočetní modely lidského vizuálního systému. Z pohledu inženýrství je cílem vytvořit systémy, které budou schopny plnit stejné úlohy jako lidský vizuální systém. Počátky počítačového vidění se objevují na konci sedmdesátých let. Zpracování vizuálních dat je obvykle založeno na extrakci příznaků, jako je nalezení hran, významných bodů a tvarů (Krčmář, 2016).

Příklady použití počítačového vidění zahrnují:

- kontrola kvality, např. průmyslový robot,
- interakce člověka s počítačem (Kinect, PlayStation Move),
- modelování objektů nebo prostředí, např. topografické modelování,
- autonomní vozidla, např. detekce dopravních značek,
- systémy rozpoznání obličeje (Face ID),
- extrakce informací pro diagnostiku pacienta, např. detekce nádorů.

Klasickým problémem v oblasti počítačového vidění je zjištění, zda data obsahují nějaký konkrétní objekt, rys nebo činnost.

2.1 ALGORITMY ROZPOZNÁNÍ OBRAZU

Tato kapitola se bude zabývat různými algoritmy počítačového vidění.

2.1.1 Předzpracování obrazu

Tato metoda slouží ke zlepšení kvality obrazu a je důležitým prvkem z hlediska dalšího zpracování. Je provedena normalizace kontrastu a jasu. K potlačení šumu je nejčastěji použita metoda filtrování průměrováním, kdy výsledná hodnota jasu pixelu je rovna průměru okolních pixelů. Rovnici číslo 2.1 zobrazuje příklad filtrovací masky

$$\boldsymbol{H} = \begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \tag{2.1}$$

kde $\boldsymbol{H}$ je filtrovací maska pro okolí 3 x 3.

Nevýhodou použití průměrování je rozmazání hran v obraze. Při zpracování barevných snímků je vhodná transformace barevného prostoru na černobílý prostor.

2.1.2 Detekce hran v obraze

Vstupní snímek má mnoho informací, které nejsou pro klasifikace nutné. Pro vnímání obrazu potřebujeme znát hrany a ostatní informace můžeme zanedbat. Výhodou je, že ubude celkové množství informací, které se dále prozkoumávají.

Každá hrana ve snímku představuje jasovou nespojitost. Tyto místa přestavují vysoký výskyt prostorových frekvencí. Oblast vysokých prostorových frekvencí přestavuje také šum, proto při jakékoliv metodě zvýrazňující hrany dojde i k zvýraznění šumu. Hrana je dána tím, jak náhle se mění hodnota obrazové funkce *f (x, y)*. K určení velikosti a směru změny hodnoty jasové funkce se používá matematická operace gradient (Horák; Kalová; Petyovský; Richter, 2008)

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f(x,\ y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,\ y)}{\partial y}\right)^2}, \tag{2.2}$$

$$\varphi = \arg\left(\frac{\partial f(x,\ y)}{\partial x}, \frac{\partial f(x,\ y)}{\partial y}\right), \tag{2.3}$$

kde $f(x, y)$ je obrazová funkce,

$\frac{\partial f(x,\ y)}{\partial x}$ – parciální derivace v zadaném bodě,

φ – úhel směru největší změny jasové funkce, °.

Výsledkem je vektorová veličina obsahující absolutní hodnotu (udává velikost změny jasové funkce) a úhel změny.

Pro detekci hran se používají gradientní operátory. Gradientní operátory aproximují derivaci obrazové funkce (Horák; Kalová; Petyovský; Richter, 2008). Mezi nejznámější konvoluční operáty patří:

- Robertsův operátor – používá okolí 2 x 2 aktuálního pixelu.

$$\boldsymbol{H} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}; \boldsymbol{H} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{2.4}$$

- Operátor Prewittové – používá okolí 3 x 3 pro osm směrů. Rovnice 2.5 zobrazuje pouze první dva směry. Ostatní je možné získat pootočením.

$$\boldsymbol{H} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}; \boldsymbol{H} = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}; \ldots \quad (2.5)$$

- Sobelův operátor – používá se převážně pro detekci vodorovných a svislých hran.

$$\boldsymbol{H} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}; \boldsymbol{H} = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix}; \ldots \quad (2.6)$$

- Robinsonův operátor.

$$\boldsymbol{H} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{pmatrix}; \boldsymbol{H} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{pmatrix}; \ldots \quad (2.7)$$

- Kirschův operátor.

$$\boldsymbol{H} = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{pmatrix}; \boldsymbol{H} = \begin{pmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{pmatrix}; \ldots \quad (2.8)$$

Cannyho hranový detektor

Jedná se o velmi populární metodu, která slouží k detekci hran ve dvourozměrném obraze. Optimální detektor musí splňovat základní tři kritéria, mezi tyto kritéria patří:

- Kritérium minimální chybovosti požaduje, aby významné hrany nebyly přehlédnuty.
- Kritérium lokalizace požaduje, aby rozdíl mezi skutečnou a nalezenou polohou hrany byl minimální.
- Kritérium jednoznačné odezvy požaduje, aby detektor reagoval na odezvu jedné hrany pouze jednou.

K dodržení těchto kritérií je Cannyho metoda realizována v několika krocích, mezi které patří (Krčmář, 2016):

1. Odstranění šumu (většinou Gaussovým filtrem).
2. Určení gradientu (nejčastěji Sobbelovým operátorem nebo první derivací Gaussovské funkce).

3. Nalezeni lokálních maxim ze zjištěných gradientů.
4. Eliminace nevýznamných hran pomocí prahování.

2.1.3 Matematická morfologie

Jedná se o metodu zpracování obrazu, která vychází z vlastností bodových množin. Obvykle pracuje s binárním obrazem, ale lze ji aplikovat i na obraz s více úrovněmi šedi. Základním morfologickým nástrojem je strukturní element, s jeho pomocí vyšetřujeme vlastnosti zkoumaného objektu. Morfologické operace se používají pro předzpracování obrazu (potlačení šumu, vyhlazení hran), zdůraznění struktur a pro kvantitativní analýzu obrazu (počet, rozměry).

Mezi základní morfologické operace patří dilatace a eroze. Dilatace skládá body dvou množin pomocí vektorového součtu. Dilatace binárního obrazu vhodnou maskou je sjednocení množiny obrazu s množinou všech posunů masky (Hájovský; Pustková; Kutálek, 2012)

$$X \oplus B = \{d \in E^2 : d = x + b, x \in X, b \in B\}. \tag{2.9}$$

Eroze kombinuje dvě množiny použitím vektorového rozdílu. Eroze binárního obrazu definována vztahem 2.10

$$X \ominus B = \{d \in E^2 : d + b \in X \ pro \ \forall b \in B\}. \tag{2.10}$$

kde X je množina obrazu

B – množina masky,

d – je bod v obraze,

E^2 – binární obrazový prostor.

2.1.4 Houghova transformace

Houghova transformace je metoda, která se používá k nalezení parametrů geometrických tvarů v obraze. Je potřeba znát analytický popis tvaru hledaného objektu. Tato metoda se používá k detekci jednoduchého známého tvaru (přímka, kružnice, elipsa, trojúhelník). Hlavní výhodou této metody je robustnost vůči nepravidelnosti hledané křivky. Tato metoda našla uplatnění například v autonomních vozidlech, kde se používala k detekci čar na vozovce.

Transformace je založena na skutečnosti, že každou přímku v rovině x, y lze jednoznačně popsat dvojicí parametrů (r, θ), tedy vzdáleností od středu souřadnicového systému a úhlem od osy x. Pro detekci přímek se používá parametrický zápis přímky v polárních souřadnicích

$$r = x \cdot \cos\theta + y \cdot \sin\theta, \tag{2.11}$$

kde r je délka přímky, m,

θ – úhel svírající přímka r s osou x, °,

x, y – kartézské souřadnice bodu.

Jestliže do rovnice dosadíme souřadnice některého bodu (x_i, y_i), pak množina možných řešení vytvoří v Houghově prostoru spojitou křivku. Pokud promítneme do Houghova prostoru všechny body ležící na nějaké přímce p, pak křivky odpovídající jednotlivým bodům (x_i, y_i) se protnou v jediném bodě (r_{max}, θ_{max}). Tato dvojice hodnot odpovídá hledaným parametrům přímky p (Horák; Kalová; Petyovský; Richter, 2008).

2.1.5 Segmentace obrazu

Cílem segmentace je rozdělit obraz na části, které souvisí s objekty z reálného světa. Takovéto části jsou tvořeny skupinou vzájemně podobných pixelů, které jsou pro další zpracování vnímány jako související oblast. Segmentace je jeden z nejdůležitějších kroků analýzy obrazu.

Prahování

Prahování šedotónového obrazu je nejjednodušší a nejstarší metoda segmentace. Díky své rychlosti a výpočetní nenáročnosti je stále hojně využívaná. Tato metoda spočívá na skutečnosti, že objekty jsou charakterizovány konstantní odrazivostí či pohltivostí svého povrchu. Pomocí určité jasové konstanty prahu převedeme obraz do binární podoby. Výsledkem bude oddělení objektů od pozadí. Proces prahování transformuje vstupní obraz na výstupní binární obraz podle vztahu (Hájovský; Pustková; Kutálek, 2012)

$$f(i,j) = \begin{cases} 1 \text{ pro } g(i,j) \geq T \\ 0 \text{ pro } g(i,j) < T \end{cases}, \tag{2.12}$$

kde T je hodnota prahu.

Správná volba prahu je naprosto kritická pro úspěšnou segmentaci objektu. Příklad níže ilustruje použití tohoto algoritmu. V prvním kroku je originální obr. 2.1 převeden na šedotónové barvy, výsledek je zobrazen na obr. 2.2. Poté je aplikován algoritmus prahování s hodnotou prahu 20. Z obr. 2.3 je patrné, že tato hodnota prahu je nedostačující. Proto na obr. 2.4 je zobrazena změna hodnoty prahu na 100. Tato hodnota nám úspěšně oddělila pozadí od objektu.

Obr. 2.1 – Originální snímek

Obr. 2.3 – Snímek převeden na šedotónové barvy

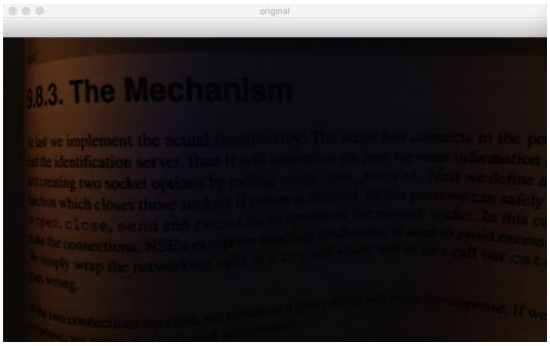
Obr. 2.4 – Prahování s hodnotou 20

Obr. 2.2 – Prahování s hodnotou 100

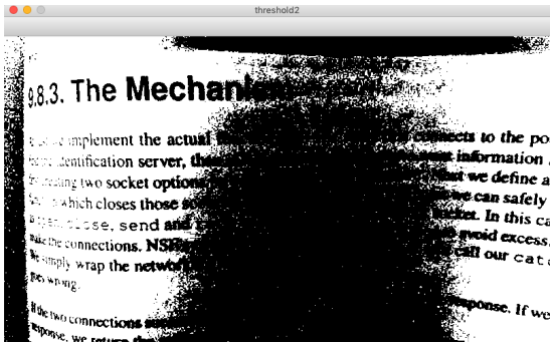
Adaptivní prahování

V této metodě prahování se hodnota prahu liší pro různé části obrazu. Obraz je nejprve rozdělen do několika částí a pro tyto části je zvolena různá hodnota prahu. Předpokladem správného užití tohoto algoritmu je správná volba velikosti části obrazu. Část musí být dostatečně velká, aby zahrnovala pixely objektů i pozadí. Při špatné volbě velikosti části dojde k chybnému výpočtu prahu, jelikož není možné určit hranici objektu a pozadí. Volba jednotlivých prahů se provádí pomocí matematických operací průměrování, mediánu, nebo průměru minimální a maximální hodnoty jasu dané části.

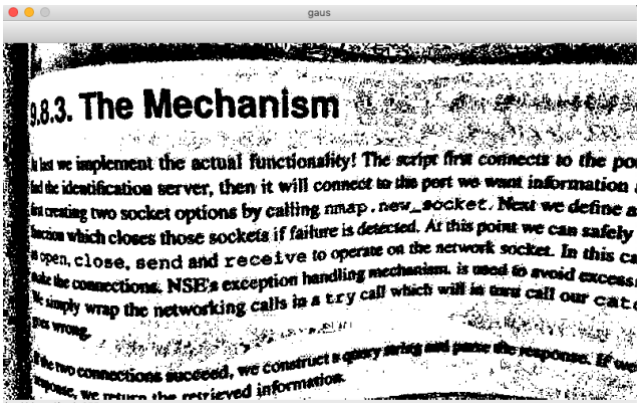
Na obr. 2.6 je zobrazen výsledek použití klasického prahování na originální obr. 2.5. Obr. 2.7 zobrazuje výsledek použití algoritmu adaptivního prahování.



Obr. 2.7 – Originální snímek



Obr. 2.6 – Požití klasického prahování



Obr. 2.5 – Použití adaptivního prahování

3 RASPBERRY PI

Jedná se o jednočipový počítač velikostí kreditní karty. Raspberry Pi bylo poprvé představeno nadací Raspberry Pi Foundation v roce 2012 s cílem podpořit výuku informačních technologií na školách. Raspberry Pi má všechny důležité porty pro připojení periferních zařízení, proto může fungovat jako klasický stolní počítač. Na desce můžeme nalézt HDMI vývod pro připojení monitoru. USB porty, přes které je možné připojit například klávesnici a myš. Reproduktory se dají připojit pomocí 3,5 mm audio konektoru. Dále na desce nalezneme slot na SD kartu, která se chová jako klasické počítačové úložiště. Raspberry Pi je oproti klasickému počítači vybaven GPIO kontakty, které mu dovolují řídit různé externí zařízení.

3.1 ZÁKLADNÍ MODELY RASPBERRY PI

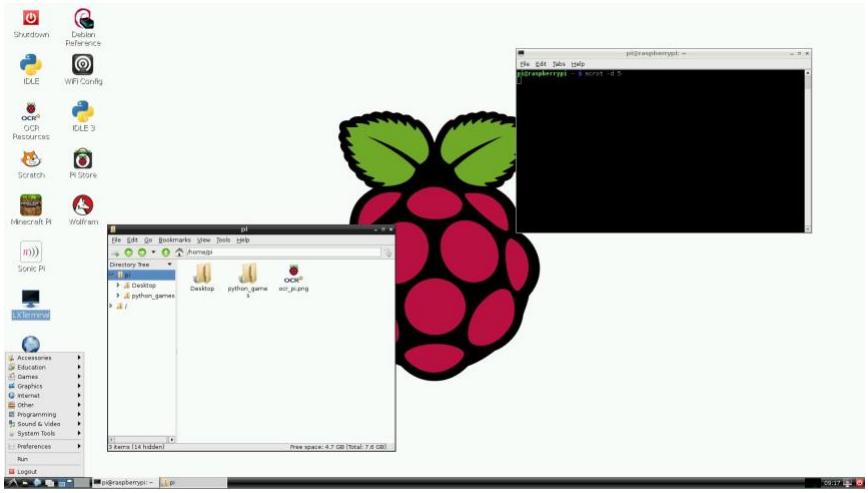
Raspberry Pi Foundation přestavila již mnoho modelů, které se liší jak cenou, tak zamýšleným použitím. V tab. 3.1 je přehled některých modelů. K realizaci praktické části je použit starší model B.

Tab. 3.1 – Přehled vybraných modelů (Wikipedia, 2019)

Model	Raspberry Pi A	Raspberry Pi 3 A+	Raspberry Pi B	Raspberry Zero v 1.2
Rok představení modelu	2013	2018	2012	2015
Frekvence procesoru	700 MHz	1,4 GHz	700 MHz	1 GHz
Velikost RAM paměti	256 MB	512 MB	512 MB	512 MB
Typ úložiště	SD/MMC	MicroSDHC	SD/MMC	MicroSDHC
Počet USB portů	1	1	2	1 x mikro USB
Počet GPIO pinů	8	17	8	17

3.2 OPERAČNÍ SYSTÉM

Oficiální operační systém podporovaný Raspberry Pi Foundation je Raspbian. Raspbian je založen na linuxové distribuci Debian a je plně optimalizován na hardware Raspberry Pi. Dodává se s plným grafickým uživatelským rozhraním a celou řadou nainstalovaných softwarů, včetně Pythonu, Scratch a Wolfram Mathematica. Raspbian byl dokončen roku 2012 Mikem Thompsonem a Peterem Greenem jako nezávislý projekt. Mezi dalšími operačními systémy umožňující chod Raspberry Pi je například Ubuntu MATE, Windows 10 loT Core a Pidora.



Obr. 3.1 – Grafické uživatelské prostředí distribuce Raspbian

3.3 PROGRAMOVACÍ JAZYKY A KNIHOVNY

Existuje značný počet programovacích jazyků, které byly přizpůsobeny pro Raspberry Pi. Raspberry Pi Foudation doporučuje programovací jazyk Python, hlavně kvůli jeho výkonosti a jednoduchosti. V podstatě každý programovací jazyk, který lze kompilovat pro rodinu procesorů ARMv6, může běžet na Raspberry Pi. Proto uživatelé nejsou omezeni pouze na používání Pythonu. Na Raspberry Pi jsou předinstalovány několik jazyků, například C, C++, Java, Scratch a Ruby.

3.3.1 Python

Python je vysokoúrovňový skriptovací jazyk pro obecné účely. Navrhl jej Guido van Rossum v roce 1991. Cílem Pythonu je poskytnout jasnou a srozumitelnou syntaxi s důrazem na čitelnost kódu. Podporuje více programovacích paradigmat, včetně objektově

orientovaného, imperativního, procedurálního nebo funkcionálního. Python je vyvíjen jako open source projekt a je dostupný pro všechny běžné operační systémy (Linux, Windows a Mac OS). V červenci roku 2018 odstoupil Guido van Rossum jako lídr open source komunity. V současné době se na trhu nacházejí dvě spolu nekompatibilní verze 2 a 3. Python 3 byl vydán v roce 2008 a odstraňuje řadu nedostatků a chybných návrhů jazyka, které nešlo odstranit, a přitom zachovat komptabilitu s nižšími verzemi. V roce 2017 bylo oznámeno, že Python 2 přestane být podporován vývojáři v roce 2020.

Ke správě balíčků využívá Python balíčkovací systém Pip. Vyšší verze než Python 2.7.9 a Python 3.4 obsahují pip ve výchozím stavu. Pro starší verze Pythonu je potřeba nainstalovat Pip pomocí příkazu

sudo apt-get install python3-pip

nebo pro verzi Python 2,

sudo apt-get install python-pip.

3.3.2 Flask

Flask je webový framework[1] napsaný v jazyce Python. Považuje se za mikroframework, protože nevyžaduje zvláštní nástroje nebo knihovny. Je navržen tak, aby umožňoval rychlé a snadné použití i pro začínající programátory. Jedná se o velice malý framework, který v základu postrádá plno funkcí (např. databázovou vrstvu nebo validaci formulářů). Flask umožňuje přidávat rozšíření a tím rozšiřovat funkce samotným aplikacím. Například existuje rozšíření pro objektově relační mapování, validaci formulářů nebo ověřování identity uživatele. Framework Flask se dá nainstalovat pomocí příkazu:

sudo pip install Flask

3.3.3 NumPy

NumPy je základní matematická Python knihovna, která se využívá pro práci s numerickými daty. NumPy poskytuje nástroje pro práci s maticemi nebo obecně s vícerozměrnými poli. Mezi tyto nástroje patří metody lineární algebry nebo Fourierovy transformace.

[1] Framework je softwarový rámec, který slouží jako podpora při programování.

NumPy je licencován pod licencí BSD[2] jako svobodný software. Předchůdce Numpy je Numeric, který byl vytvořen v roce 1995. Jméno NumPy se používá od roku 2006, kdy vývojář Trevis Oliphant sloučil knihovnu Numeric s konkurenční knihovnou Numarray.

3.3.4 OpenCV

OpenCV je svobodná a otevřená multiplatformní knihovna pro práci s počítačovým viděním a se strojovým učením. Cílem OpenCV je vytvořit společnou infrastrukturu pro aplikace počítačového vidění a tím urychlit vývoj celého odvětví. Knihovna je plně zdarma pod licencí BSD s otevřeným zdrojovým kódem. Knihovna obsahuje celou řadu nejmodernějších algoritmů pro práci s počítačovým viděním. Tyto algoritmy mohou být použity například k detekci a rozpoznání obličeje, identifikaci objektů, sledování pohybů objektů, 3D modelů objektů, najití podobných snímků z databáze snímků nebo odstranění červených očí. Knihovna je široce využívána ve firmách, výzkumných skupinách a vládních orgánech. OpenCV je dostupný pro Windows, Linux, Android a Mac OS. OpenCV je napsán v jazyce C++. Jeho primární rozhraní je v jazyce C++, ale je i dostupný v Pythonu, Javě a Matlabu/Octave.

3.4 SMTP PROTOKOL

SMTP je internetový protokol určený pro přenos emailových zpráv. Protokol doručuje poštu pomocí navázaného spojení mezi odesílatelem a adresátem. Účastníci přenosu používají protokol SMTP na portu TCP/25. Protokol SMTP se stará o doručení zprávy do emailové schránky adresáta, který k ní pak může kdykoliv přistupovat pomocí protokolů IMAP a POP3.

Tento protokol je v praktické části práce použit pro odeslání emailu uživateli, pokud dojde k otevření dveří nebo zaznamenání osoby na kamerovém modulu.

3.5 GPIO KONEKTORY

GPIO konektory umožňují připojit elektronický hardware k Raspberry Pi, například diody, relé, motory, senzory, snímače a mnoho dalších. Každý pin lze ovládat pomocí programovacího jazyka, jako je například Python. Provozní napětí Raspberry Pi je +5 V, ale GPIO piny mohou být vystaveny maximálnímu napětí +3,3 V. Konektory mohou být použity jako vstupy nebo výstupy pro všeobecné použití. Existují dva stavy při použití GPIO pinu jako

[2] BSD licence je jedna z nejsvobodnější z licencí. Umožňuje šíření obsahu, přičemž vyžaduje pouze uvedení autora.

vstupu a to logická 1, nebo 0. Tyto stavy jsou popsány jako úrovně napětí. Napětí vyšší než 1,8 V odpovídá stavu 1 a napětí pod 1,8 V odpovídá stavu 0. Existují také dva stavy při definování GPIO pinu jako výstupu. Když je pin v logickém stavu 1, znamená to, že úroveň napětí odpovídá 3,3 V. Oproti tomu, logický stav 0 popisuje napětí 0 V.

Konektory GPIO se v různých modelech Raspberry Pi liší. Rozložení konektorů v modelu Raspberry Pi B je zobrazeno v tab. 3.2.

Tab. 3.2 – Rozložení pinů v Raspberry Pi
(Wikipedia, 2019)

funkce	Pin	Pin	funkce
+3,3 V	1	2	+5 V
SDA1	3	4	+5 V
SLC1	5	6	GND
GCLK	7	8	TXD0 (UART)
GND	9	10	RXD0 (UART)
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
+3,3 V	17	18	GPIO 24
MOSI (SPI)	19	20	GND
MISO (SPI)	21	22	GPIO 25
SCLK (SPI)	23	24	CE0_N (SPI)
GND	25	26	CE0_N (SPI)

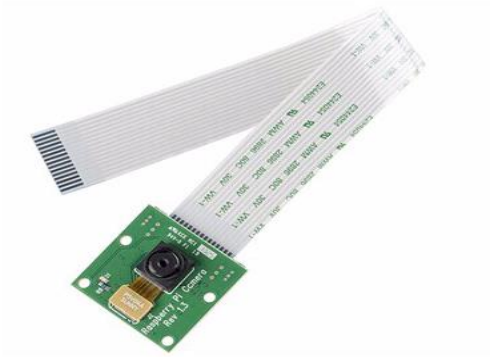
3.6 BEZDRÁTOVÝ WI-FI ADAPTÉR

K realizaci praktické části je použit model Raspberry Pi B, který nemá Wi-Fi rozhraní. Z tohoto důvodu je k Raspberry Pi připojen USB Wi-Fi adaptér, který zajišťuje bezdrátové připojení. Adaptér značky Edimax je kompatibilní s bezdrátovými standardy 802.11x a dosahuje přenosové rychlosti až 150 Mbps. Adaptér podporuje speciální verzi WLAN, která inteligentně upravuje výstup přenosu podle vzdálenost, čímž pomáhá snížit spotřebu energie při bezdrátovém provozu. Takto lze snížit spotřebu energie až o 20 % ~ 50 % (Edimax Technology, 2014).

Obr. 3.1 –
Bezdrátový Wi-Fi adaptér
(Edimax Technology, 2014)

3.7 KAMEROVÝ MODUL

Jak již bylo zmíněno dříve, existuje několik komponentů pro Raspberry Pi, které lze zakoupit samostatně. Jeden z těchto doplňků je kamerový modul s názvem Raspberry Pi Camera board. Modul má rozlišení 8 Mpx (3280 x 2464 pixelů). Kamerový modul lze připojit k Raspberry Pi pomocí rozhraní CSI. Toto rozhraní se ve velké míře používá i u fotoaparátů v mobilních telefonech, pro které bylo původně navrženo. Komunikace je vedena pouze jednosměrně (z modulu do mikropočítače). Rozhraní CSI je schopno dosahovat vysokých přenosových rychlostí a je speciálně navrženo pro přenášení informací o pixelech. Velikost desky nesoucí modul je 24 mm x 25 mm x 9 mm a váží kolem 3 g, proto se hodí pro mobilní aplikace, kde velikost a váha jsou důležité parametry. Konektor se na desce nachází mezi HDMI a Ethernet portem (Jenč, 2014).



Obr. 3.2 – Kamerový modul
(Greetech Wiki, 2014)

3.8 MAGNETICKÝ DETEKTOR POHYBU

Pro rozpoznání otevření dveří je v této práci použit magnetický drátový senzor pohybu. Výhodou magnetických senzorů je to, že magnetické pole prochází i nemagnetickými materiály. Senzory dokáží rozpoznávat magnety, kterou jsou umístěny například za překážkami z barevných kovů, hliníku, dřeva nebo umělých hmot. Detektor nevyžaduje žádné samostatné napájení. Signál je vyslán do řídicí jednotky při oddálení dvou magneticky propojených částí o více než 2 cm. Při rozevření jednoho dílu na rámu a druhého na pohyblivé části dojde k rozepnutí kontaktu a je vyslán signál.

Tento typ senzoru je založen na tak zvaném Reedově principu. Tento princip byl vynalezen v roce 1936. Základem senzoru jsou dva feromagnetické jazýčky uložené ve skleněné baňce s inertním plynem. V klidovém stavu jsou jazýčky rozepnuty, nepůsobí na ně žádné magnetické pole. Po přidáním magnetického pole dojde ke sepnutí jazýčků. Elektrická indukce vyvolaná přiloženým magnetickým polem vytvoří ve feromagnetickém materiálu opačnou magnetickou polaritu, která svými protikladnými póly přitáhne jazýčky a způsobí vodivé spojení kontaktů. Odebrání magnetického pole způsobí navrácení jazýčků do klidového stavu (Vojáček, 2017).

Obr. 3.3 – Magnetický senzor
pohybu (iGet Security, 2019)

4 NÁVRH A IMPLEMENTACE ZAŘÍZENÍ

4.1 SSH SPOJENÍ

Komunikační protokol SSH umožňuje vzdálený přístup z jednoho počítače do druhého. K využití této služby je potřeba mít nainstalovaného SSH klienta. Na počítačích s operačními systémy Linux nebo macOS je SSH klient součástí operačního systému. Na počítačích s operačním systémem Microsoft Windows je potřeba stáhnout nástroj např. Putty, který zpřístupní síťový protokol SSH.

K aktivaci SSH je potřeba znát IP adresu Raspberry Pi, ta se dá zjistit například pomocí příkazu

ifconfig | grep “inet addr:”,

metoda grep vybere pouze řádky, které obsahují výraz specifikovaný v uvozovkách.

```
pi@raspberrypi ~ $ ifconfig | grep "inet addr:"
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet addr:192.168.0.123  Bcast:192.168.0.255  Mask:255.255.255.0
```

Obr. 4.1 – Výstup příkazu ifconfig

Z obr. 4.1 je patrné, že IP adresa je 192.168.0.123. Poté už stačí tuto IP adresu vložit do příkazu

ssh pi@192.168.0.123.

Po spuštění tohoto příkazu se program zeptá na přihlašovací heslo. Po zadaní správného hesla se na počítači zpřístupní Raspberry Pi terminál.

4.2 NASTAVENÍ EXTERNÍHO WI-FI ADAPTÉRU

Raspberry Pi k bezdrátovému připojení k internetu využívá externí Wi-Fi adaptér. K tomu, aby bylo Raspberry Pi správně připojeno k místní Wi-Fi síti je potřeba mít nastaveno několik věcí:

1. Prvně je potřeba zjistit, jestli je adaptér správně připojen k mikropočítači. K tomu složí příkaz lsusb, který zobrazí na konzoli všechny zařízení připojené k USB portům. Výstup onoho příkazu je zobrazen na obr. 4.2. Ze čtvrtého řádku je patrné, že mikropočítač úspěšně rozpoznal externí adaptér.

```
pi@raspberrypi ~ $ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 004: ID 7392:7811 Edimax Technology Co., Ltd EW-7811Un 802.11n Wireless Adapter [Realtek RTL8188CUS]
```

Obr. 4.2 – Výstup příkazu lsusb

2. Dále je potřeba zkontrolovat, jestli jsou ovladače zařízení načteny. Pro zobrazení jádrem načtených modulů lze použít příkaz lsmod. Tento příkaz zobrazí list modulů, ve kterém musí být zapsán modul cfg80211. Jedná se konfigurační rozhraní pro zařízení využívající standardu 802.11[3] v systému Linux.
3. V posledním kroku stačí otevřít soubor /etc/wpa_supplicant/wpa_supplicant.conf a v něm specifikovat údaje o lokální Wi-Fi síti. Konfigurační soubor může obsahovat takovéto nastavení:

```
network = {
    ssid="JMENO"
    proto=RSN
    key_mgmt=WPA-PSK
    pairwise=CCMP TKIP
    group=CCMP TKIP
    psk="HESLO"
},
```

kde ssid obsahuje jméno lokální sítě. Parametr proto specifikuje seznam přijatelných protokolů. V tomto případě RSN znamená, že je povolen pouze protokol WPA2. Key_mgmt specifikuje protokol správy klíčů. Parametr pairwise a group se stará o správné šifrování. Psk obsahuje heslo sítě.

4.3 INSTALACE KNIHOVNY OPENCV NA RASPBERRY PI

V této kapitole je vysvětlen postup instalace OpenCV 3.0 pro Python 2.7 na mikropočítač Raspberry Pi model B. Podrobněji se čtenář může seznámit s průběhem instalace (Rosebrock, 2015). Celková instalace knihovny trvala necelých 8 hodin. To je dáno tím, že Raspberry Pi model B disponuje poměrně starými hardwarovými komponenty.

V prvním kroku je potřeba stáhnout a nainstalovat vývojářské nástroje potřebné pro instalaci knihovny

sudo apt-get install build-essential cmake git.

[3] Standard 802.11 definuje normu pro lokální bezdrátové sítě.

V druhém kroku stačí stáhnout knihovnu OpenCV 3.1.0 z oficiálního GitHub[4] repositáře. K tomuto účelu lze použít nástroj git, který byl stažen a nainstalován v minulém příkazu

git clone https://github.com/Itseez/opencv.git.

4.3.1 Nastavení virtuálního prostředí

Pro vytvoření samostatného virtuálního prostředí slouží balíčky virtualenv a virtualenvwapper. Instalace těchto balíčků není podmínkou k instalaci samotné knihovny OpenCV, avšak je to standardní vývojářská praktika

sudo pip install virtualenv virtualenvwrapper.

Nyní je potřeba aktualizovat soubor /.profile. V tomto souboru lze specifikovat příkazy, které se automaticky spustí po přihlášení uživatele do systému. Přidáním těchto řádků do souboru se docílí správného nastavení virtuálního prostředí:

```
export VIRTUALENVWRAPPER_PYTHON = /usr/bin/python2.7
export WORKON_HOME=$HOME /.virtualenvs
source /usr/local/bin/virtualenvwrapper.sh.
```

První řádek specifikuje umístění souboru, který dokáže interpretovat kód napsaný v Pythonu (tzv. Python interpret). Druhý řádek nastaví aktuální domovský adresář jako místo, kde mají být uloženy virtuální prostředí. Poslední příkaz spouští samotný program virtualenvwrapper.sh.

Nyní lze vytvořit nové virtuální prostředí příkazem

mkvirtualenv cv3,

kde cv3 je jméno nového virtuálního prostředí. Pro připojení do nově vytvořeného virtuálního prostředí se používá příkaz

workon cv3.

4.3.2 Kompilace OpenCV

OpenCV závisí na knihovně NumPy, proto dříve než se předstoupí k samotné kompilaci, je potřeba nainstalovat tuto knihovnu. Knihovna se nainstaluje příkazem

sudo pip install numpy.

[4] GitHub je služba nabízející bezplatný Webhosting pro open source projekty.

K řízení procesu kompilace se využívá multiplatformní nástroj CMake. Pro účel kompilace musí být vytvořena složka uvnitř staženého repositáře, ve které budou uloženy kompilované soubory.

```
cd ~/opencv
mkdir build
cd build
```

Předtím, než je samotná kompilace zahájena, je potřeba spustit příkaz, který specifikuje jednotlivé parametry kompilace:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
      -D CMAKE_INSTALL_PREFIX=/usr/local \
      -D INSTALL_PYTHON_EXAMPLES=ON \
      -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
      -D BUILD_EXAMPLES=ON ..
```

První řádek nastaví typ sestavy. Příkaz CMAKE_INSTALL_PREFIX nastaví cestu, kde mají být uloženy nainstalované soubory. Třetí řádek je volitelný a stará se o instalaci vzorových příkladů kódů psaných v jazyku Python pro knihovnu OpenCV. Čtvrtý řádek specifikuje cestu pro instalaci dodatečných modulů. Poslední řádek sestaví všechny příklady. Dvě tečky poukazují na nadřazený adresář, kde se nachází soubor CMakeLists.txt, který je nezbytný pro kompilaci.

Nyní stačí spustit proces kompilace příkazem

```
make –j4.
```

Přepínač –j nastaví počet jader. Použití více jader dramaticky urychlí kompilaci, i přes to, proces kompilace trvá několik hodin.

4.3.3 Instalace OpenCV

Pokud kompilace proběhla bez chyby, může se předstoupit k samotné instalaci. Instalace se vyvolá příkazem:

```
sudo make install
```

K ověření, zda je knihovna správně nainstalována, poslouží Python interpret, kterého lze spustit jednoduchým příkazem python. Druhém kroku se načte knihovna OpenCV pomocí příkazu,

```
import cv2.
```

Pokud by v této fázi nebyla knihovna dobře nainstalovaná, zobrazila by se chybová hláška. Pro zobrazení verze knihovny OpenCV lze použít příkaz:

```
cv2.__version__.
```

Jestliže výsledkem příkazu je číslo verze, znamená to, že instalace knihovny OpenCV proběhla úspěšně.

Z obr. 4.3 plyne, že na Raspberry Pi je nainstalována verze OpenCV 3.1.0.

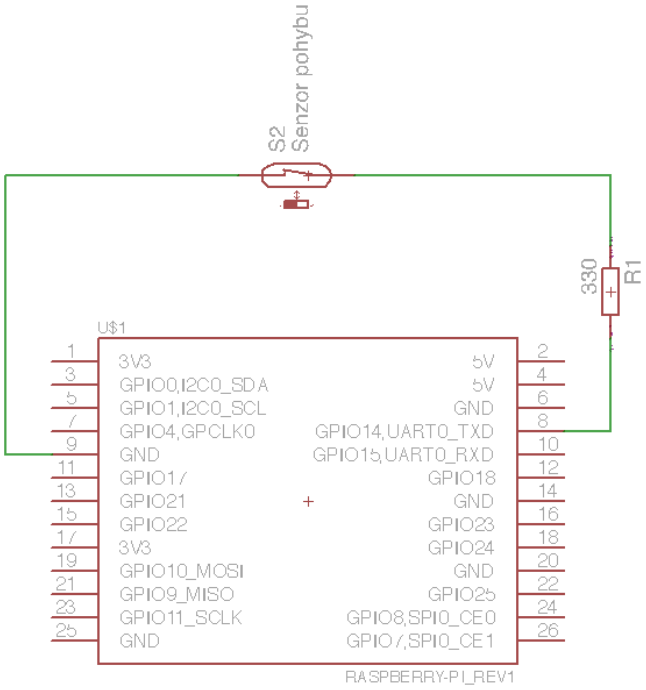
```
(cv3) pi@raspberrypi ~/Smart-Security-Camera $ python
Python 2.7.3 (default, Nov 24 2017, 21:13:24)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.1.0'
>>>
```

Obr. 4.3 – Ověření, zda instalace proběhla v pořádku

4.4 HARDWAROVÉ ŘEŠENÍ

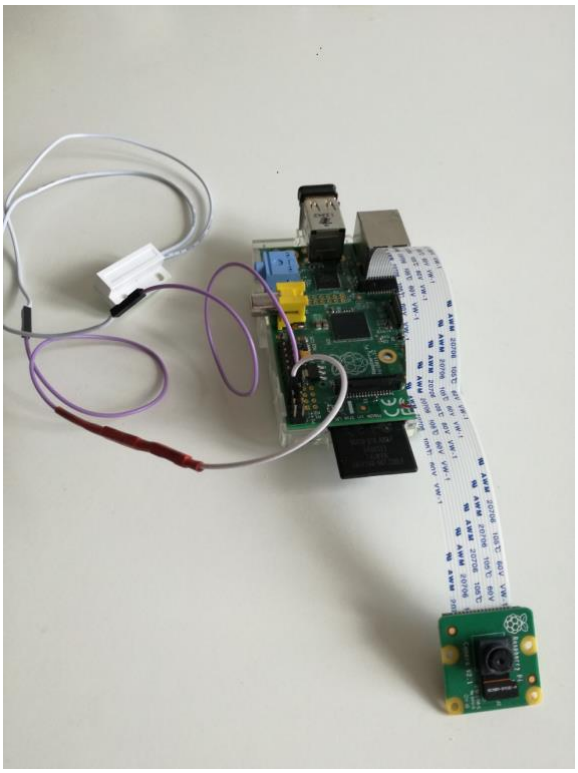
Jelikož kamerový modul je přímo vyrobený pro Raspberry Pi, stačí ho pouze připojit pomocí flexového kabelu k rozhraní CSI.

Magnetický senzor pohybu byl k desce připojen pomocí konektorů číslo 8 a 9. Konektor číslo 8 je roven GPIO pinu číslo 14 a konektor číslo 9 je zemnící pin (viz tab. 4.2). Pro zvýšení bezpečnosti mezi pinem GPIO14 a detektorem pohybu byl vložen odpor o velikosti 330 Ω.



Obr. 4.4 – Zapojení senzoru pohybu

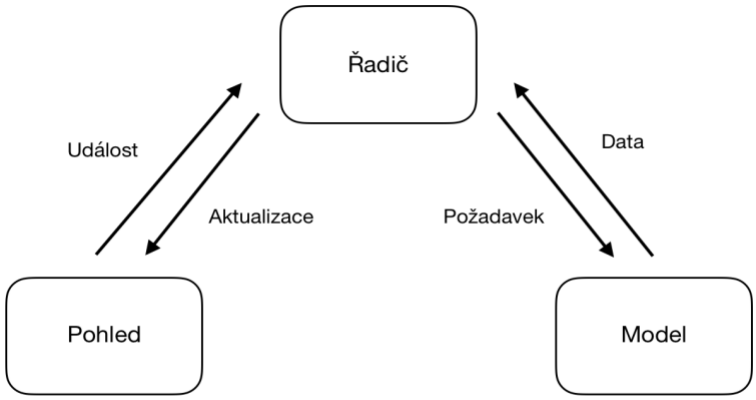
Na obr. 4.5 je zobrazen snímek Raspberry Pi k němuž je připojen detektor pohybu a kamerový modul.



Obr. 4.5 – Fotka realizovaného zařízení

4.5 STRUKTURA APLIKACE

Architektura aplikace odpovídá softwarovému návrhovému vzoru MVC (Model-view-controller). MVC architektura odděluje datový model aplikace (model), řídicí logiku (controller) a uživatelské rozhraní (view) do tří nezávislých komponent tak, že modifikace některé z nich má jen minimální vliv na ostatní. Takto napsaná aplikace má větší přehlednost, udržitelnost a škálovatelnost, než použití jen jednoho souboru pro celý projekt (tvz. „špagetový kód“). Na obr. číslo 4.6 je vyobrazeno grafické schéma standartní MVC architektury.



Obr. 4.6 – Grafické schéma MVC architektury

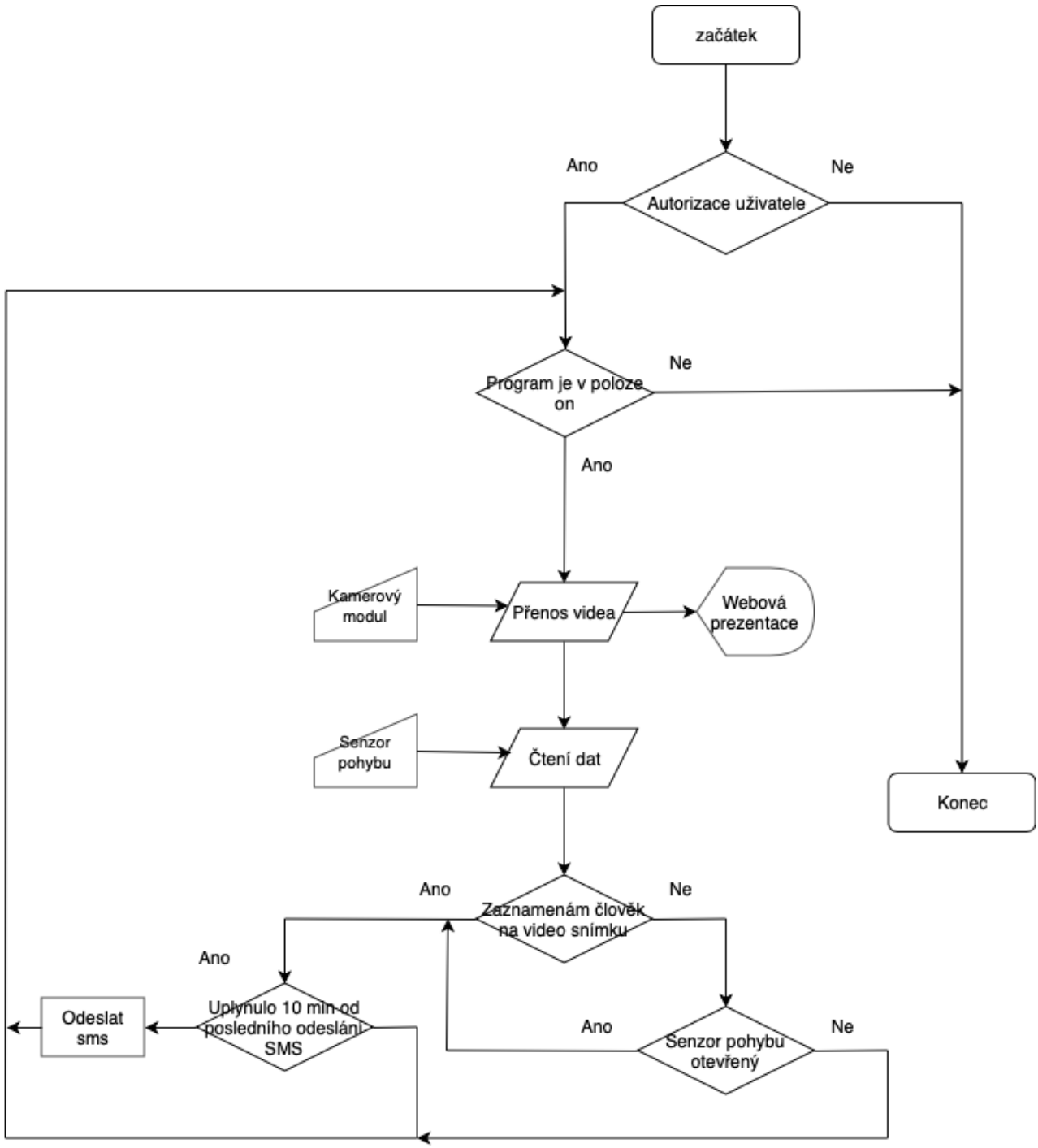
Aplikace obsahuje několik souborů, které jsou rozděleny do těchto tří komponentů.

- Soubory camera.py, move.py a mail.py jsou modely aplikace (M). Starají se pouze o vnitřní logiku. Například rozeznání člověka na kameře nebo rozeznání otevření dveří a tuto informaci předají řídicí části aplikace.
- Soubor index.html se stará o správnou webovou prezentaci (V).
- Soubor main.py je řídicí část aplikace (C), která reaguje na události přicházející z modelů. Jedná se o jakéhosi prostředníka, který komunikuje se všemi ostatními komponentami aplikace.

Na obr. číslo 4.7 je zobrazena adresářová struktura aplikace.

Obr. 4.7 – Adresářová struktura aplikace

4.6 VÝVOJOVÝ DIAGRAM



Obr. 4.8 – Vývojový diagram

4.7 MODEL ROZPOZNÁNÍ ČLOVĚKA

Pro rozpoznání člověka na video snímku je použit model lidského těla, který vytvořili vědci z univerzity ETH Zurich. Dohromady publikovali tři rozdílné lidské modely:

- model horní části těla,
- model rozpoznání obličeje,
- model rozpoznání celého těla.

Všechny tyto modely podporují pouze čelní a zadní pohledy. Boční pohled je složitější, a aby byl úspěšný, musel by například zahrnovat informaci o pohybu. V práci je použit model horní části těla, který po provedení testů je pro tento účel nejvhodnější.

Model rozpoznání obličeje je nepřesnější model, protože vzor je zřetelně odlišený od ostatních tvarů vyskytující se v pozadí. Tak tomu není pro model horní části těla nebo zejména pro model celého těla, protože vzor se musí spoléhat na křehké informace o siluetě spíše než o detaily. Všechny modely jsou psané ve značkovacím jazyku XML.

Model je načten v souboru main.py

object_classifier = cv2.CascadeClassifier("detectors/upperbody_recognition_model.xml").

V tomto řádku kódu je načten požadovaný XML klasifikátor a uložen do proměnné object_classifier. V souboru camera.py je deklarována metoda get_object s parametrem object_classifier, která je volána v řídicím programu main.py. Obsah této metody je zobrazen na obr. 4.9.

```
def get_object(self, classifier):
    found_objects = False
    frame = self.flip_if_needed(self.vs.read()).copy()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    objects = classifier.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),
        flags=cv2.CASCADE_SCALE_IMAGE
    )

    if len(objects) > 0:
        found_objects = True

    for (x, y, w, h) in objects:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

    ret, jpeg = cv2.imencode('.jpg', frame)
    return (jpeg.tobytes(), found_objects)
```

Obr. 4.9 – Obsah metody get_object

V této metodě je důležitá funkce detectMultiScale, která je součástí knihovny OpenCV. Tato funkce se stará o správnou detekci objektů ve vstupním snímku.

Prvním parametrem je samotný snímek, který je převeden na šedotónové barvy. Druhý parametr udává, o kolik procent se změní velikost měřítka vstupního snímku. Tímto parametrem se docílí, že i objekty s jinou velikostí, než na které byl model natrénovaný, budou algoritmem nalezeny. Třetí parametr určuje kvalitu detekce. Vyšší hodnota parametru vede k menší četnosti detekce s vyšší kvalitou. Čtvrtý parametr určuje minimální velikost objektu. Objekty menší, než určená hodnota jsou ignorovány.

Další část funkce se stará o správné formátování výsledku. Kolem nalezeného objektu je nakreslen zelený obdélník a uložen jako nový snímek. Poté je tento snímek převeden na .jpeg a registrován jako výstup funkce.

4.8 ROZPOZNÁNÍ OTEVŘENÍ DVEŘÍ

Soubor move.py má za úkol hlídat stav senzoru pohybu. V první části programu je importována knihovna RPi.GPIO, která slouží k ovládání GPIO pinů. Poté je nutné definovat, jaké číslování konektorů bude v projektu zvoleno. Volba GPIO.BOARD určuje, že v projektu je odkazováno na čísla pinů, a ne na čísla jednotlivých GPIO konektorů (viz tab. 3.2).

Ke čtení stavu pinu slouží metoda get_door_state. Metoda čte stav pinu číslo 8, kde je připojen senzor pohybu. Jestliže je stav roven binární hodnotě 1, znamená to, že mezi pinem a zemí je napětí 3,3 V. Tento stav je způsoben rozepnutím senzoru pohybu. Metoda je volána hlavním programu main.py, kde podle zjištěného stavu proběhne odeslání emailu.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(8, GPIO.IN, pull_up_down=GPIO.PUD_UP)

class MoveDetector():
        def get_door_state(self):
                if GPIO.input(8):
                        state=True
                else:
                        state=False
                return state
```

Obr. 4.10 – Obsah souboru move.py

4.9 ODESLÁNÍ EMAILU

Pro správné odeslání emailu se stará soubor mail.py. Soubor využívá dvě python knihovny, smtlib a email. Knihovna smtlib obsahuje klienta SMTP služby. Tento klient pomocí SMTP démona dokáže odeslat poštu na libovolné zařízení, které je připojené k internetu. Zabudovaný emailový balíček v jazyku Python umožňuje přenášet složité emailové zprávy. Dnešní nejběžnější typ e-mailu je MIME kombinující HTML a prostý text. Zprávy MIME jsou zpracovávány Python modulem email.MIME. Obsah souboru mail.py je zobrazen na obr. 4.11.

```
import smtplib
from email.MIMEMultipart import MIMEMultipart
from email.MIMEText import MIMEText
from email.MIMEImage import MIMEImage

fromEmail = 'raspberrycameraproject@gmail.com'
fromEmailPassword = 'password123'

toEmail = 'st49522@student.upce.cz'

def sendEmail(image=None):
        msgRoot = MIMEMultipart('related')
        msgRoot['Subject'] = 'Security Update'
        msgRoot['From'] = fromEmail
        msgRoot['To'] = toEmail
        msgRoot.preamble = 'Raspberry pi security camera update'

        msgAlternative = MIMEMultipart('alternative')
        msgRoot.attach(msgAlternative)
        msgText = MIMEText('Your doors are open!')
        msgAlternative.attach(msgText)

        if image is not None:
                msgText = MIMEText('<img src="cid:image1">', 'html')
                msgAlternative.attach(msgText)

                msgImage = MIMEImage(image)
                msgImage.add_header('Content-ID', '<image1>')
                msgRoot.attach(msgImage)

                msgText = MIMEText('Smart security cam found object')
                msgAlternative.attach(msgText)

        smtp = smtplib.SMTP('smtp.gmail.com', 587)
        smtp.starttls()
        smtp.login(fromEmail, fromEmailPassword)
        smtp.sendmail(fromEmail, toEmail, msgRoot.as_string())
        smtp.quit()
```

Obr. 4.11 – Obsah souboru mail.py

V horní části kódu je importován SMTP klient a jednotlivé moduly z baličku email, které dovolují přidávat emailu různé funkce.

V druhé části kódu je deklarována funkce sendEmail s parametrem image. Tuto funkci volá v souboru main.py jak kamerový modul, tak detektor pohybu. Pokud je funkce volána kamerovým modulem, je za parametr image vložen obrázek nalezeného objektu. Pokud funkci

volá detektor pohybu, je parametr nulový („None“). V další části kódu jsou k emailu přidány různé atributy, jako je například odesílatel, adresát, předmět a text zprávy. Jestliže obrázek nalezeného objektu je dostupný, je tento obrázek také přidán do obsahu zprávy.

V konečné části kódu je definována služba Gmail jako SMTP server s portem 587. Nad definovanou službou proběhne přihlášení a odeslání připraveného emailu.

4.10 ŘÍDICÍ PROGRAM

Program starají se o správné ovládání aplikace je uložen v souboru main.py. Tento program obstarává nejen správnou webovou prezentaci a autorizaci uživatele, ale také reaguje na události, které přicházejí ze souborů camera.py a move.py. Mezi tyto události patří rozpoznání objektu na kamerovém snímku nebo rozpoznání otevření dveří. Funkce, která sleduje stavy přicházející ze souborů, se jmenuje check_for_objects a je zobrazena na obr 4.12.

```
def check_for_objects():
        global last_epoch
        while True:
                try:
                        frame, found_obj = video_camera.get_object(object_classifier)
                        move_state = move_detector.get_door_state()
                        if found_obj and (time.time() - last_epoch) > email_update_interval:
                                last_epoch = time.time()
                                print "Sending email..."
                                sendEmail(frame)
                                print "done!"
                        if move_state and (time.time() - last_epoch) > email_update_interval:
                                last_epoch = time.time()
                                print "Sending email..."
                                sendEmail()
                                print "done!"
                except:
                        print "Error sending email: ", sys.exc_info()[0]
```

Obr. 4.12 – Řídicí funkce

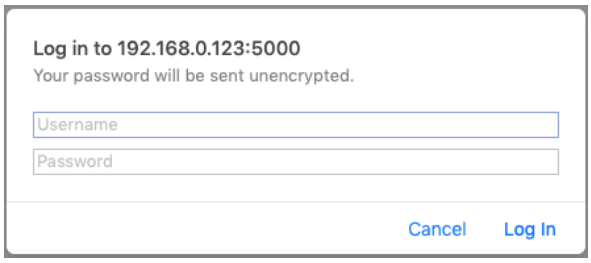
Tato funkce je zabalená do nekonečné smyčky a pouze přerušení z terminálu ji může ukončit. Takové chování je žádoucí, protože se očekává, že program bude sledovat kamerový záznam a detektor pohybu nepřetržitě.

V první části programu se volá klasifikátor objektu, který při úspěšném nalezení objektu vrací snímek nalezeného člověka a logickou hodnotu true. Dříve, než je snímek odeslán adresátovi, je zkontrolováno, jestli uběhlo 10 minut od posledního odeslaného emailu. Časový interval je tam proto, aby adresát nebyl příliš zatěžován stejnými emaily. Stejný proces se

opakuje i pro detektor pohybu, jen s tím rozdílem, že metoda move_state vrací pouze logickou hodnotu true nebo false.

4.11 WEBOVÁ PREZENTACE

Živý záznam přenáší Raspberry Pi na adresu 192.168.0.123:5000, která se skládá z IP adresy mikropočítače a portu. Port 5000 používá Flask jako výchozí port pro vývojářské účely. Pro zobrazení živého záznamu je nutné, aby zařízení, ze kterého se klient připojuje, bylo ve stejné síti jako mikropočítač. Pokud je tato podmínka splněna, stačí vložit tuto adresu do webového prohlížeče. Tento krok aktivuje jednoduchou autorizaci, která je nastavena v souboru main.py. Jedná se o bezpečnostní pojistku, která bez autorizačních údajů nedovolí uživateli vstoupit do aplikace. Na obr. 4.13 je zobrazeno okno autorizačního procesu.

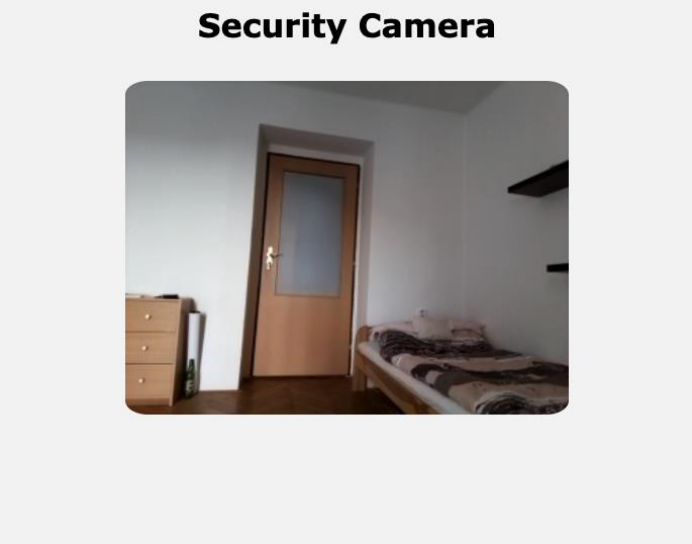


Obr. 4.13 – Autorizační proces

Autorizační údaje jsou staticky uloženy v souboru main.py.

```
app.config['BASIC_AUTH_USERNAME'] = 'username'
app.config['BASIC_AUTH_PASSWORD'] = 'password'
```

Úspěšný autorizační proces přesměruje uživatele na kořenový uzel, kde se vygeneruje html stránka, index.html. V tomto kroku už je hlavní program spuštěn a stránka ukazuje živý obraz z kamerového modulu.



Obr. 4.14 – Kořenová stránka index.html

Kód, který se stará o autorizaci a generování kořenové stránky, je zobrazen níže.

```
@app.route('/')
@basic_auth.required
def index():
    return render_template('index.html')
```

Obr. 4.15 – Generování kořenové stránky

Soubor index.html obsahuje základní html štítky a kaskádové styly. Kaskádové styly se starají o správné zarovnání, velikosti a obtékání jednotlivých elementů na webové stránce.

Nejzajímavější řádek v souboru index.html je

<img id="bg" src="{{ url_for('video_feed') }}">.

Tento kód se stará o zobrazení živého záznamu. Jedná se o snímek, který je definovaný štítkem <img>. Zdroj onoho snímku je obsah stránky /video_feed. Funkce url_for transformuje funkci video_feed na URL stránku, která je definována v souboru main.py.

4.12 NASTAVENÍ AUTOMATICKÉHO SPUŠTĚNÍ APLIKACE

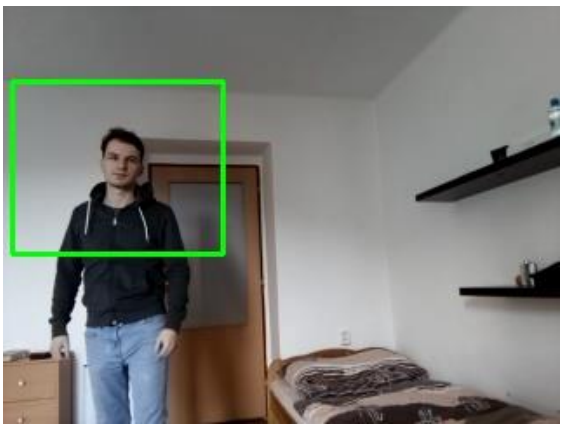
Aby se docílilo automatického spuštění programu při zapnutí Raspberry Pi, je potřeba nakonfigurovat soubor /etc/rc.local. Uvnitř souboru se specifikují příkazy, které mikropočítač vykoná hned po načtení operačního systému. Přidáním příkazu pro spuštění hlavního souboru main.py se docílí automatického spuštění aplikace. Obsah souboru je zobrazen na obr 4.16.

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

python /home/pi/SecuritySystem/main.py

exit 0
```

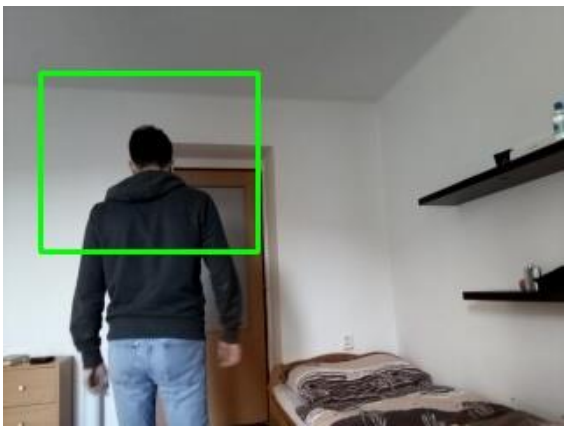
Obr. 4.16 – Obsah souboru /etc/rc.local

4.13 VYHODNOCENÍ VÝSLEDKŮ

V této kapitole je vyhodnocena správnost nalezení objektu na video snímku. Správnost nalezení byla testována podle snímku, na kterém klasifikátor nalezl objekt obsahující horní část lidského těla a tento snímek odeslal na zadaný email. Na obr. 4.17 a obr. 4.18 je demonstrováno úspěšné nalezení objektu.



Obr. 4.17 – Výsledný snímek č. 1

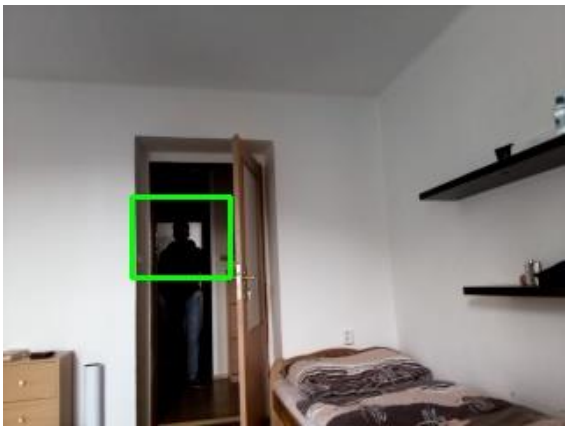


Obr. 4.18 – Výsledný snímek č. 2

Jak je ze snímků patrné, klasifikátor byl schopný nalézt jak čelní pohled horní části těla, tak i zadní pohled.

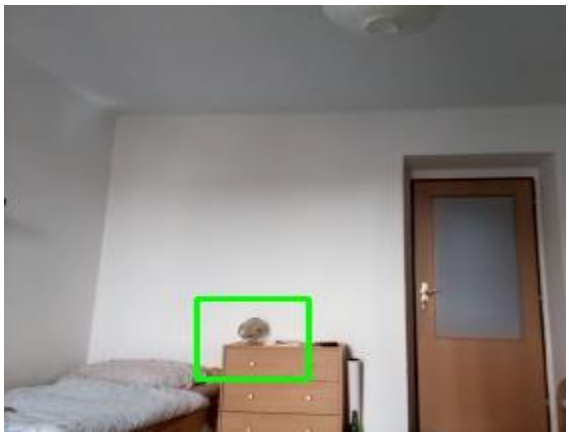
Člověk, zobrazený na obr. 4.17, vešel do místnosti dveřmi, ale klasifikátor ho rozpoznal až poté, co zavřel dveře a narovnal se čelně ke kameře. Zde se potvrdilo pravidlo, které je napsané v kapitole 4.7, a to že klasifikátor umí rozpoznávat pouze čelní a zadní pohledy.

Na obr. 4.19 je zajímavé sledovat, že klasifikátor byl schopný správně vyhodnotit vzdálený objekt, který postrádá detaily. To je dáno tím, že klasifikátor neposuzuje detaily obličeje, ale opírá se pouze o siluetu objektu.



Obr. 4.19 – Výsledný snímek č. 3

Opírat se pouze o siluetu objektu může znamenat více chybných vyhodnocení. Na obr 4.20 je zobrazeno malé zrcadlo, které klasifikátor také vyhodnotil jako horní část lidského těla. Zrcadlo může tvarem připomínat lidskou hlavu a roh skříňky znázorňovat lidské rameno.



Obr. 4.20 – Výsledný snímek č. 4

Úspěšnost systému byla testována na objektech, které klasifikátor vyhodnotil jako horní část lidského těla. Celkem bylo hodnoceno 35 případů. Z tohoto vzorku bylo 28 případů správně vyhodnoceno a 7 chybně. Tudíž výsledná úspěšnost systému je stanovena na 80 %. Chybně hodnocené případy většinou obsahují předměty, které tvarem připomínají horní část lidského těla, jako je například zrcadlo z obr. 4.20.

5 ZÁVĚR

Cílem práce bylo realizovat zabezpečovací systém na bázi Raspberry Pi. K realizaci systému se schopností rozeznat člověka byla použita knihovna pro počítačové vidění OpenCV. Pomocí této knihovny bylo ukázáno, jak relativně jednoduše může počítač vyřešit složitý úkol, a to rozeznat člověka od okolních objektů. V současné době roste zájem o tuto vědní oblast, například technologičtí giganti jako je Google a Facebook nedávno zveřejnily otevřené knihovny Tensorflow a PyTorch pro práci s počítačovým viděním. Počítačové vidění hraje důležitou roli v nadcházejícím konceptu čtvrté průmyslové revoluce. Metody počítačového vidění budou schopné nahradit člověka v jeho schopnosti vidět a viděného objektu rozumět.

Přesnost klasifikace v této práci byla stanovena na 80 %. Ta by se dala zvýšit, pokud by se nastavil minimální práh velikosti objektu a objekty menší by byly z klasifikace úplně zanedbány. Tato změna by zvýšila přesnost klasifikace z blízkosti, ale snížila by klasifikační dosah systému. Celkové náklady na bezpečností systém jsou cca 1700 Kč a je funkcemi srovnatelný s komerčními bezpečnostními systémy od 2000 Kč. Výhodou vytvořeného systému je možnost libovolné změny v provedení nebo vylepšení stávajícího nastavení. Nevýhoda systému je, že nemá možnost nastavení známých lidí, které nemá klasifikátor vyhodnotit jako hrozbu a následně pro tyto osoby vypnout sledovač pohybu, který upozorňuje na každé otevření dveří. Přidání takového vylepšení by bylo složité, protože použitý klasifikátor se spoléhá pouze na obrysy horní části těla a nezajímá se o jednotlivé detaily. Jednoduší řešení by bylo použít klasifikátor obličeje, protože obličej osahuje více různorodých detailů, na které se může klasifikátor odkazovat. V takovém to případě by systém musel snímat obličeje z blízkosti, aby zachytil všechny potřebné detaily.

POUŽITÁ LITERATURA

CCTV Cameras Explained [online]. London: Techcude Limited, 2016 [cit. 2019-01-7]. Dostupné z: https://www.techcube.co.uk/blog/cctv-cameras-explained/

EDIMAX TECHNOLOGY. Wireless Adapter EW-7811Un, 2014 [online] [cit. 2019-02-16]. Dostupné z: https://www.edimax.com/edimax/merchandise/merchandise_detail/data/edimax/in/wireless_adapters_n150/ew-7811un/

HÁJOVSKÝ, R.; PUSTKOVÁ, R.; KUTÁLEK, F.; *Zpracování obrazu v měřící a řídicí technice*, Ostrava: Technická univerzita Ostrava, 2012 [online] [cit. 2018-12-27]. Dostupné z: http://www.person.vsb.cz/archivcd/FEI/ZOMRT/Zpracovani%20obrazu%20v%20merici%20a%20ridici%20technice.pdf

HORÁK, K.; KALOVÁ, I.; PETYOVSKÝ, P.; RICHTER, M. *Počítačové vidění* [online]. Brno: Vysoké učení technické v Brně, 2008 [cit. 2019-02-17]. Dostupné z: http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/Pocitacove_videni_S.pdf

iGet Security. [online]. Brno: Intelek spol.s.r.o., 2019 [cit. 2019-02-07]. Dostupné z: http://www.iget.eu/cs/p8-security-magneticky-dratovy-detektor-dvere-okna

JENČ, J. *Možnost rozšíření Raspberry Pi o modul kamery* [online]. Praha: České vysoké učení technické v Praze, 2014 [2018-01-04]. Dostupné z: http://users.fs.cvut.cz/ivo.bukovsky/PVVR/prace_studentu/Jenc_RPi_kamery_s_FFT.pdf

KRČMÁŘ, M. *Zpracování obrazu v zařízení Android – detekce a rozpoznání vizitky* [online]. Brno, 2016 [cit. 2018-12-16]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=124507. Práce ke státní diplomové zkoušce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřící techniky.

MINICHINO, J.; HOWSE J. 2015. *Learning OpenCV 3 Computer Vision with Python Second Edition*. Birmingham (UK): Packt Publishing Ltd. 356 p. ISBN 978-1-78528-384-0

Raspberry Pi. In: Wikipedia: The Free Encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2019-01-22] Dostupné z: https://en.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=894014011

Raspberry Pi Camera Module. In: Greeetech Wiki [online]. 2014 [cit. 2019-01-24]. Dostupné z: https://www.geeetech.com/wiki/index.php/Raspberry_Pi_Camera_Module

ROBINSON, A.; COOK, M. 2014. *Raspberry Pi: Projects*. Chichester, West Sussex (UK): John Wiley & Sons. 480 p. ISBN 978-1-118-55543-9

ROSEBROCK, A. Installing OpenCV 3.0 for both Python 2.7 and Python 3+ on your Raspberry Pi 2 [online]. 2015 [cit. 2019-02-11]. Dostupné z: https://www.pyimagesearch.com/2015/07/27/installing-opencv-3-0-for-both-python-2-7-and-python-3-on-your-raspberry-pi-2/

SOLEM, E. J. *Programming Computer Vision with Python* [online]. 2012 [cit. 2018-12-05]. Dostupné z: http://programmingcomputervision.com/downloads/ProgrammingComputerVision_CCdraft.pdf

UPTON, E.; HALFACREE, G. 2016. *Raspberry Pi User Guide, 4th Edition*. Chichester, West Sussex (UK): John Wiley & Sons. 320 p. ISBN 978-1-119-26436-1.

VOJÁČEK, A. Magnetické senzory přiblížení - 1. díl [online]. 2017 [cit. 2019-01-16]. Dostupné z: https://automatizace.hw.cz/magneticke-senzory-priblizeni.html